https://ntrs.nasa.gov/search.jsp?R=20160008899 2019-08-31T02:11:00+00:00Z

Semantic Representation and Scale-up of Integrated Air Traffic Management Data

Rich Keller, Ph.D.* Shubha Ranjan+◆
Mei Wei* Michelle Eshow◆

*Intelligent Systems Division / ◆Aviation Systems Division
+Moffett Technologies, Inc.

NASA Ames Research Center

Work funded by NASA's Aeronautics Research Mission Directorate

Point of contact:
rich.keller@nasa.gov

Aviation Data is Big Data

- **Volume**: 30M+ flights yearly
  3.6B passengers forecast for 2016
- **Variety**: flight tracks, weather maps, aircraft maintenance records, flight charts, baggage routing data, passenger itineraries
- **Velocity**: high frequency data from multiple aviation and aircraft systems for multi-hour flights

Semantic Big Data for Aviation: *Two Initial Questions*

1. Can semantic representations be used to advantage in aviation data management? (Absolutely!)
2. But can semantic representations scale to accomplish practical tasks using Big Data? (I was not so sure…)

<u>Project</u>: Build a large queryable semantic repository of historical data about the US Air Traffic Management system, and test how its performance scales up

Background: NASA's Air Traffic Management (ATM) Data Warehouse

- NASA researchers require historical ATM data
  - NASA Ames conducts research on future ATM concepts
  - Researchers require data for analysis and concept validation
- NASA Ames' **ATM Data Warehouse** archives data collected from FAA, NASA, NOAA, DOT, industry
  - Warehouse captures 13 different sources of aviation data:
    - flight track data, flight route data, weather data, flight stats
    - some from live feeds and some from periodic updates
  - Data holdings available back to 2009
  - 30TB of data

Problem: Non-integrated Data

- ATM Warehouse data is replicated & archived in its original format
- Data sets lack standardization
  - data formats
  - nomenclature
  - conceptual structure

> - **Possible cross-dataset mismatches:**
>   - terminology
>   - scientific units
>   - temporal alignment
>   - spatial alignment
>   - conceptualization organization

- To analyze and mine data, researchers must write special-purpose code to integrate data for each new task

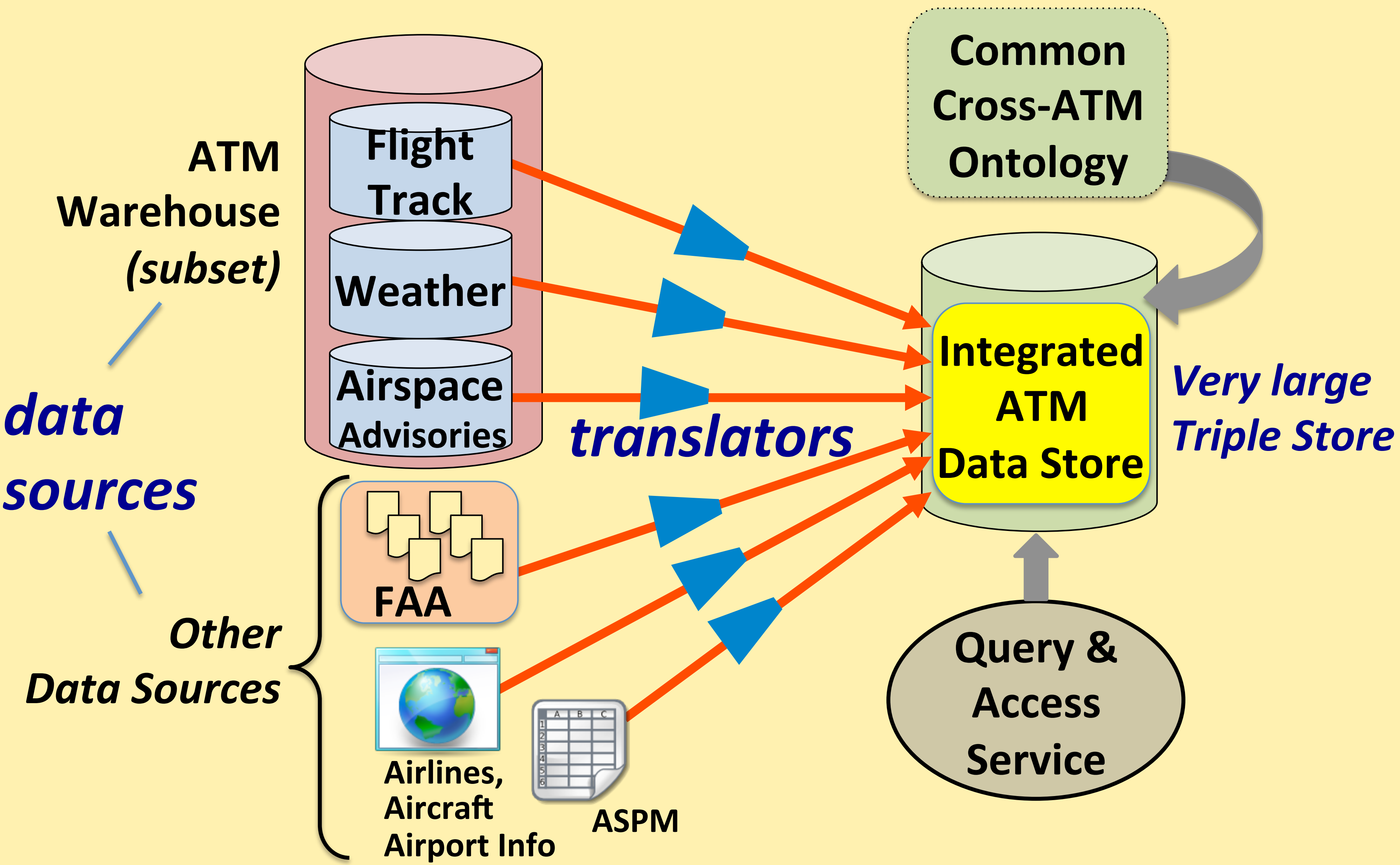
  ➔ <u>Huge time sink</u>!

Proposed Solution

Relieve users of responsibility for integration

Integrate Warehouse data sources
on the server side
using **Semantic Integration**

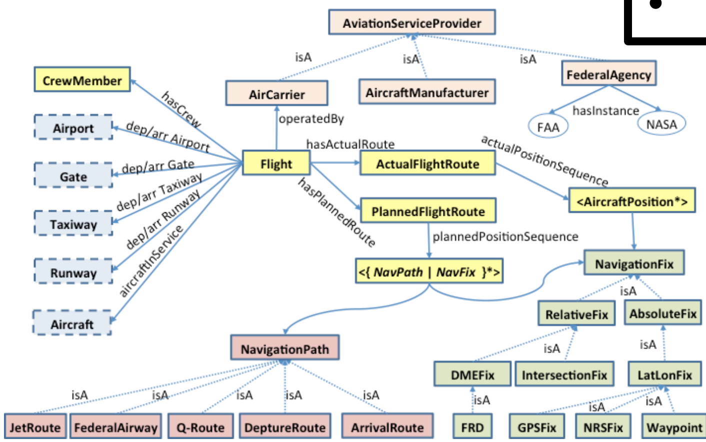
Semantic Integration Approach:
Prototype System Diagram



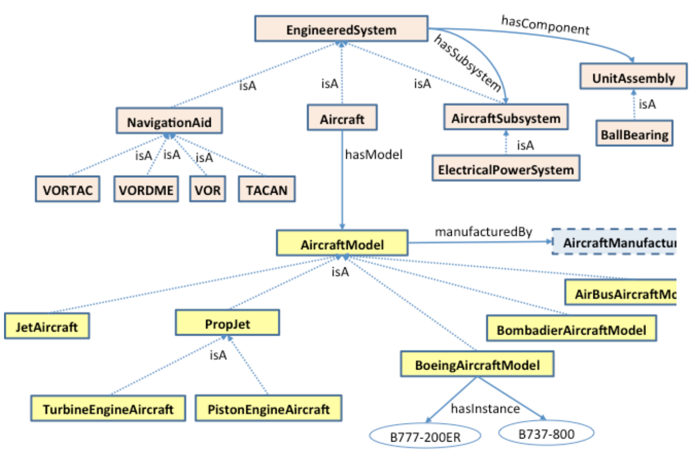
ATM Ontology

- 150+ classes
- 150+ datatype properties
- 100+ object properties

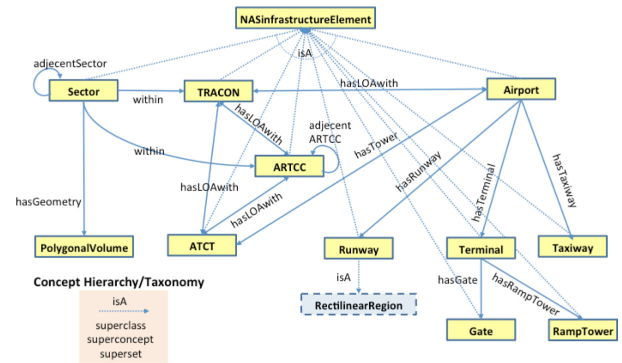
Flight & Navigation



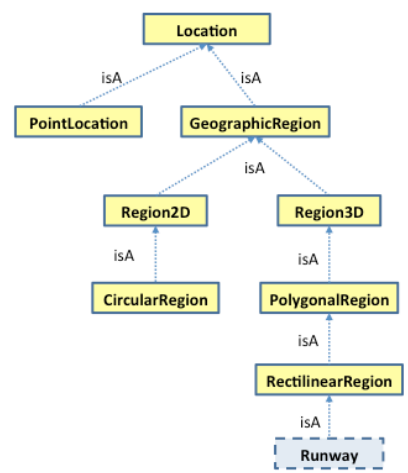
Aviation Equipment



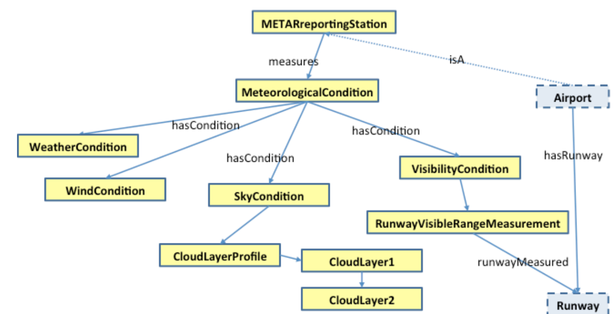
NAS Infrastructure



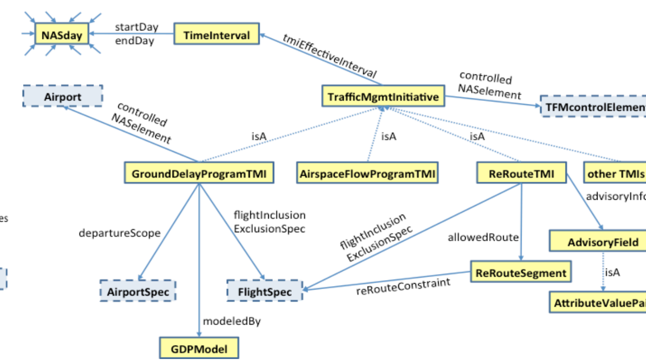
Spatial Representation



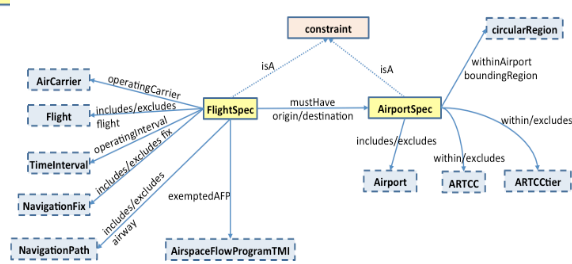
Meteorology



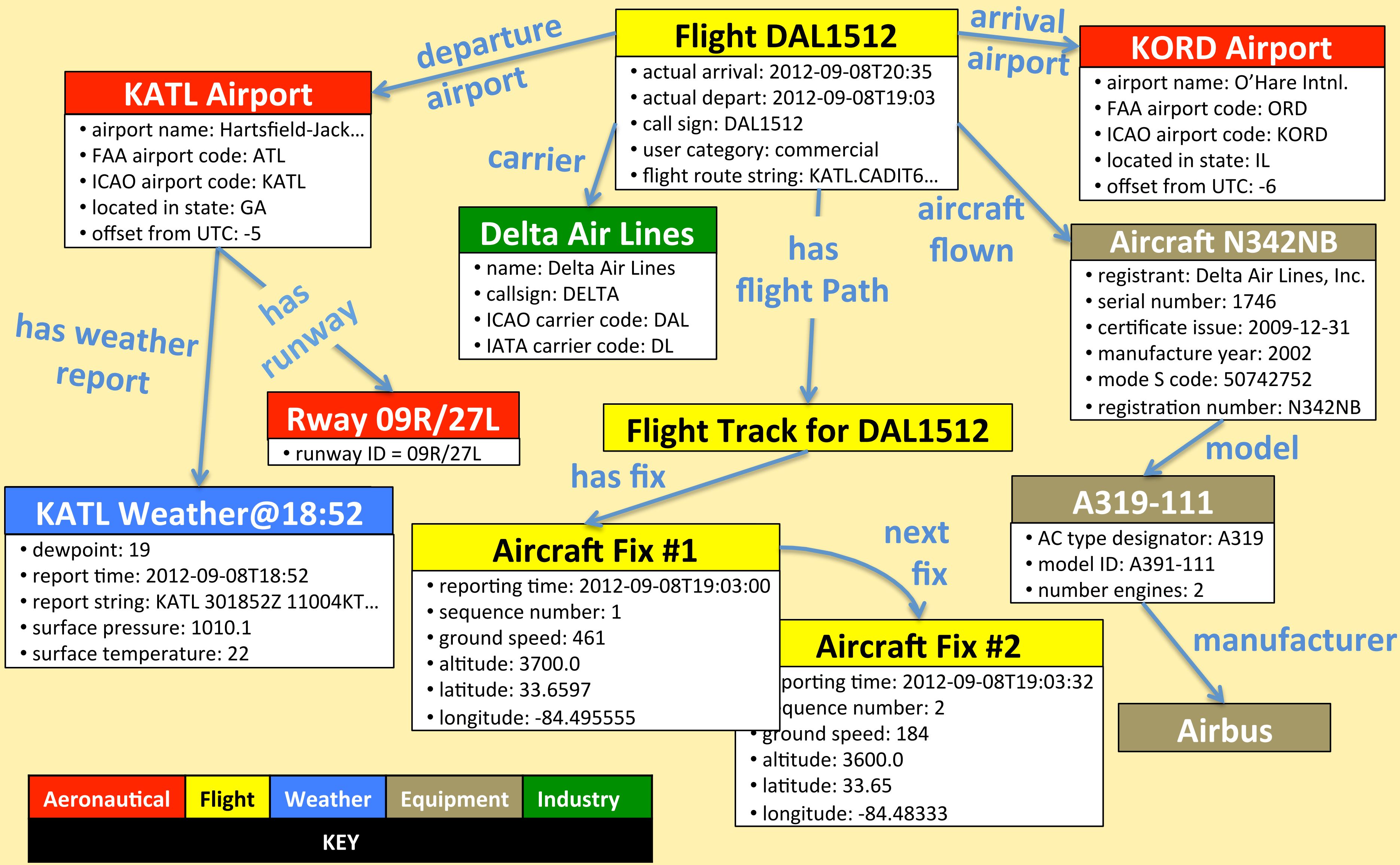
Traffic Management Initiatives (TMIs)



Flight/Airport Constraints



Ontology Representation of a Flight



Experimental Methodology

1. Develop ontology
2. Write data source translators
3. Run translators to generate/load data into triple store for a period covering 1 day of air traffic to/from a major airport (Atlanta): 1342 flights; ~2.4M triples
4. Develop and run a set of SPARQL benchmark queries against two commercial triple stores
   (AllegroGraph and GraphDB)
5. Generate and load synthetic triples scaling to one month of air traffic: ~40+K flights; ~36M triples*
6. Run queries again to compare results

*Estimate: 10B triples / year for US domestic flight traffic

Representative SPARQL Queries

from benchmark set of 17 queries for evaluating performance on scale-up

- Flight Demographics:
  - F1: Find Delta flights using A319s departing ZTL airports
  - F3: Find flights with rainy departures from ATL
- Airspace Sector Capacity:
  - S6: Find the busiest airspace sectors for a day, aggregating hourly
- FAA Traffic Management Impacts:
  - T1: Find flights that were subject to ground delays
- Weather-Impacted Traffic Index (WITI):
  - W1: Calculate hourly WITI values
- Flight Delay Data:
  - A3: Compare hourly airport acceptance rate with arrival demand

Results for 17 benchmark queries

Flight Period	Execution Time		
	Min	Max	Avg
1 Day	11 ms	9.6 sec	1.19 sec
1 Month	8 ms	1620.3 sec (27 min)	96.65 sec (1.6 min)

- ~60% of queries scaled in proportion to increase in triples
- ~30% of queries experienced no increase
- 1 query experienced exponential increase (350x – 700x, depending on triple store)

Query #	Execution Time in Milliseconds				Scale Factor	
	2.4M triples		36M triples		36M/2.4M ratio	
	Store #1	Store #2	Store #1	Store #2	Store #1	Store #2
A1	49	197	53	210	1.08	1.07
A2	36	176	37	147	1.03	0.84
A3	12	37	8	31	0.67	0.84
F1	98	64	2584	324	26.37	5.06
F2	36	28	298	96	8.28	3.43
F3S	466	482	12462	5070	26.74	10.52
S1S	1033	4749	726565	1651215	703.35	347.70
S2	11	858	59	19363	5.36	22.57
S3	1844	6060	35500	115389	19.25	19.04
S4	1786	4991	34985	108882	19.59	21.82
S5	1096	1412	11170	31199	10.19	22.10
S6	4825	9640	96846	163205	20.07	16.93
T1	32	43	269	171	8.41	3.98
T2	11	28	8	42	0.73	1.50
T3	193	68	268898	259	1393.25	3.81
W1	11	33	426	130	38.73	3.94
W2	11	37	11	39	1.00	1.05

Conclusion: Scaling to multi-year flight periods does not appear feasible *unless* multi-hour or multi-day response times are acceptable

Potential Scale-Up Approaches

- **Hardware**: triple 'appliances' for storage & processing
- **Algorithm**: better graph matching algorithms
- **Software**: better query planners; new indexing approaches
- **Query reformulation**: rewrite queries
- **Triple reduction**: reduce graph search space

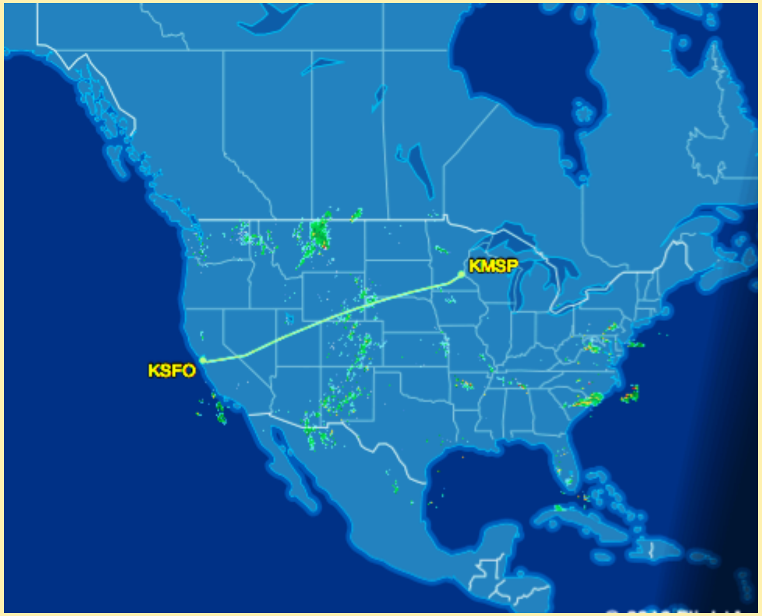
Query Reformulation

- SPARQL queries can (in theory) be rewritten to improve efficiency
- Lack of transparency regarding how SPARQL queries are translated into code and executed makes rewriting difficult
- Tools to assist with optimization are missing or poorly documented
- Wanted: query plan inspector, index formulation tools, performance monitoring
- SQL performance analysis tools are mature; SPARQL tools are primitive (in our experience)

Triple Reduction

- Reduce the underlying search space by modifying the representation
- Undesirable trade-off possible:
  → trade representational fidelity for efficiency

Example: *representation of Aircraft Track Points*



TrackPoint Representation Tradeoff

Representation #1 vs. Representation #2

(2 per minute: ~70% of all instances)

AircraftTrackPoint

- reporting time: 2012-09-08T19:03:00
- sequence number: 31
- ground speed: 461

hasFix

GeographicFix

- altitude: 3700.0
- latitude: 33.6597
- longitude: -84.495555

(1 per minute: ~54% of all instances)

AircraftTrackPoint

- reporting time: 2012-09-08T19:03:00
- sequence number: 31
- ground speed: 461
- altitude: 3700.0
- latitude: 33.6597
- longitude: -84.495555

Current Status Update

- Have scaled up to 1 month of actual flight data from the three NY Metropolitan airports:
  ~257M triples
  → considerably more than the 36M/month reported for Atlanta airport in the paper

- Will be re-testing benchmark queries against this data, but not easily comparable to existing data due to changed geographic region

Summary

- Described a real-world practical application for big semantic data: *integration of heterogeneous ATM data*
- Reviewed experiments performed to scale-up data and measure impact on query performance
- Discussed some approaches to improving performance (query reformulation and triple reduction)

<u>Conclusion</u>: Adequate tools not yet available to support real-world performance tuning for SPARQL queries in commercial triple stores

<u>Caveat</u>: Experience limited to only 2 triple stores!

In the end

Can semantic representations scale to accomplish practical tasks using Big Data?

(Well, I'm *still* not sure…
to be continued…)